# SINC - LINK

September - October 1986 Vol. 4 No. 5

SINC-LINC IS A PUBLICATION OF THE TORONTO TIMEX-SINCLAIR USERS CLUB AND IS ISSUED 6 TIMES A YEAR. COPIES OF THE NEWSLETTER ARE $1.50 EACH FOR NON-MEMBERS, MEMBERS RECEIVE A FRRE COPY AS PART OF THE $20.00 ANNUAL MEMBERSHIP FEE.

NEWSLETTERS ARE EXCHANGE FREE OF CHARGE WITH OTHER TIMEX-SINCLAIR USERS GROUPS.



SEND CORRESPONDENCE TO:
SINC-LINC
C/O SEAN WENZEL
757 VICTORIA PRK. AVE.
STE. 302
TORONTO, ONTARIO
M4C 5N8

INDEX



FOR SALE:

2 Word Processing Systems COCO compatible

32k tape based, parallel print output, $120
64k w/ 2 drives, parallel print output, $400
Add $50 for 300 baud modem
Contact Ian Singer at (416)699-6380

## THE HACKER SCARE

by John T. Nguyen

Part 1 of 3

Back in 1983 it seemed like everywhere I turned I would find an article on kids breaking into computer systems. These stories were not limited to just magazines and newspapers. Television also gave these stories a fair amount of air time. Being an avid computer enthusiast myself, I read and watched with interest at the illegal, but yet colorful activities of these kids (known as "hackers") and marvelled at how they had brought national attention to themselves. Everyone including government officials (Caspar Weinberger in particular) took notice of these teen agers and seriously looked at the threat they posed to the security of computer systems across the nation.

Groups like the 414's made headlines by breaking into the computer systems of government installations and corporate agencies. From 1982 to 1985, these hackers received a great deal of publicity from the media. They were praised, criticized, and at times, feared. The public, by and large, did not understand the crimes these teenagers committed. As a result the nation as a whole had mixed feelings of fear and doubt about computer security and about computers in general. The period from 1982 to 1985 spans this period of "electronic uneasiness" or more specifically, the period of the "hacker scare".

If anyone has read an article about these teenagers they undoubtedly have stumbled on the word "hacker", a term used to describe these kids who break into computer systems.

Traditionally, however, the word had very different meanings. About 20 years back, "hacking" was used by early programmers to refer to the process of chopping or hacking a program down in size and thus make it more efficient at each turn. When "hacking" was used as a noun (i.e. hacker), it referred to someone who was totally devoted to computers and spent every waking moment in the computer room. In the 1970's "hacker" took on a more general meaning of "someone who works with computers in one form or another, more specifically in programming". Starting in 1982, "hacking" and "hacker" took on still another meaning. Reporters who wrote stories about teenagers who broke into computer systems referred to them as "hackers". And every time "hacker" was mentioned, the connotation of "computer break-ins" was assumed.

How did the hacker craze get started in the first place? To answer this question it will be necessary to go back to the early 1960's when the first hackers first into existance. These early hackers were young computer wizards (many of whome were school drop-outs) who were employed by the artificial -intelligence and computer research laboratories at M.I.T. Because of their expertise, M.I.T. hired them to develop sophisticated software and hardware. These hackers were a strange group of people who ate and slept at strange hours, but their devotion to computers resulted in some of the most important advances in computer science. They were responsible for writing the first chess-playing programs; the first "expert system" (this was a program which was programmed to "know" a subject --such as medicine, law, or engineering, etc.-- really well and can be "asked" to answer questions in these subjects); and the first time-sharing system which made computers more accessible to programmers. They also developed the first TV-like screens for computers which are so common today. Previously programmers basically had a typewriter with continuous paper to do their programming on. As strange as it may sound, an activity which these programmers enjoyed partaking in was the deliberate "crashing" of their computer systems. This usually meant temporarily putting the machine out of order through clever means. The purpose of which was to find flaws in the system so as to correct it afterward.

In the 1970's, M.I.T. hackers were still around, but there was yet another type of hacker which grew into existance. In his book,"OUT OF THE INNER CIRCLE", Bill Langreth gives a very interesting adcount of the telephone hackers of the 70's, and their importance to the hackers of the 80's. The popular name given to these hackers at the time was phone phreak. The phreaks had a great interest in exploring the world-wide telephone network without paying the cost of making the calls. How was this possible? Around 1950 AT&T decided to base their long-distance switching system on a multi-frequency system, whereby audible tones were used to perform various functions such as opening lines, switching from local to long-distance trunks, and many other tasks which human operators used to do. Soon afterward, a blind 8-year old boy by the name of Joe Engressia discovered that by whistling at different frequencies to his phone, he was able to make it switch to a long-distance line, and he was able to make long distance calls for free. He did this for years, and in college he whistled up phone calls for his friends. Soon word got around about what Joe was doing, and others started doing it. As a result phreaks started to spring up from everything from pipe organs to flutes to tape recorders. The activities of these phone phreaks were not entirely unlike the activities of the hackers of the 80's which we will look at shortly.

In looking at the hackers from these various decades one can see similarities in all of them. They were usually young individuals who had a rebellious nature about them. They tended to stray from the confines of societal rules and regulations and can be viewed as somewhat mischievous. 1980's hackers are no different than their predecessors in this sense.

From SLUG, the St. Louisville Users Group newsletter.

retyped by GFC

## TS 2068 SOUND ROUTINES

GUNSHOTS:

```
10 SOUND 6,15;7,7;8,16;9,16;10,16;12,16;13,0
20 PAUSE 60
30 GOTO 10
```

WHISTLING BOMB:

```
10 SOUND 7,62;8,15
20 FOR I = 50 TO 100
30 SOUND 0,I; PAUSE 3
40 NEXT I
```

# BOB'S NOTEBOOK

## FONT LOADER

One of my interests is collecting character sets or fonts and storing them on a disk with the following loader.

```
   1 REM ggggbCODE 27P2A\;F\ RET
URN COPY LINE ATTR )*J\);PEEK (7
#####) REM PEEK BIN COPY n
```

(REMDOS)

```
  10)REM #0)##UDOS#
  20 OUT 34,0
  30 POKE 23606,0: POKE 23607,60
 100 INPUT "address? (eg,51200)"
;"add
 110 RANDOMIZE add: POKE 23540,P
EEK 23670: POKE 23541,PEEK 23671
 120 LET lsb=PEEK 23670: LET msb
=PEEK 23671
 125 PRINT " lsb= ";lsb,"msb= ";m
sb
 130 PAUSE 120
 210 CLS : INPUT "name? (eg, hea
vy2.C1)"n$: RANDOMIZE USR 26720
: REM move n$
 220 POKE 23606,lsb: POKE 23607,
msb-1
 290 CLS
 300 FOR i=32 TO 255
 310 PRINT CHR$ i;
 320 NEXT i
 330 STOP
 400 RANDOMIZE USR 26720: REM sa
ve "fontld.BI
 410 RUN
```

At line 1 is the visible part of REMDOS, a disk version of the Larken DOS which I merged into the program from tape. These disk DOS's allow for extra commands and the use of variable names when saving, etc. One of the extra commands is "move" which allows code to be loaded to a different address than the one it was originally saved to.

So, the loader makes use of these features to allow you to select a font by name from your disk directory and then load it into memory starting at any address you choose.

After you input your choice of start address at line 100, it is RANDOMIZED, storing the lsb and msb automatically in addresses 23670/1. These are POKED into addresses 23540/1 for later use with the "move" command.

The lsb and msb values appear on the screen briefly and you should write them down.

You are then prompted for the name of the font you wish to load. Enter it exactly as it appears on the directory with no quotes.

This set is then moved to the new address and then addresses 23606/7 are POKED with values lsb and msb-1.

When you want to use the loader, make sure you pick a RAM area which will not clash with the BASIC program or any other m/c you are going to load in.

When your programs are entered, you will have to re-POKE 23606/7 with the values lsb and msb-1.

Because the Sinclair font is so light, it is best replaced by a heavier set in any program where you usually make a 2040 copy. This may be done in other ways than the one described above. For example, you could "move" the code to a suitable address and save it to disk, then "load" it along with your program each time you use it. Or you could put the code into DATA statements and POKE it to the proper address each time you load your program.

**********************************

In the Aug/Sep issue, my item about the ON ERR routine for the Spectrum has an error (mine) in line 8 of column 2. It should read (obviously) POKE 23610,255.

**********************************

In Rename (same issue), I later found that an EI (enable interrupt routine) can be more easily accomplished by:

RANDOMIZE USR 31

so let ei=31 and leave out line 20. This ROM call is the same for both TS2068 and Spectrum. In case you are interested I used my EDITOR program (copy in club tape library) to search for the string F8C9 (the enable interrupt bytes). The search took a second or so to locate three locations in the entire ROM.

BOB MITCHELL SEP 86

## TRS-80 TO ZX81 CONVERSIONS

**by George Chambers**
**from W.J. Henry material**

CONVERSION FOR READ/DATA:

**Original TRS-80 sample program to be converted.**

```
1000 DATA 1,4,28,123,730,4040
1010 DIM G(7)
1020 FOR J =1 TO 7
1030 READ G(J)
1040 NEXT J
```

To convert we store the information (data) in a string

Sample TS1000 conversion.

```
1000 LET C$ = "000100040060012307304040"
                    (note the zero groups)
1020 DIM G(J)       (sets up array)
1030 FOR J = 1 TO 7  (picks out the data items from
                                the data groups)
1040 LET G(J) = VAL A$ (4*J-33 TO 4*J)
              (the key to the whole operation)
1050 NEXT J      (gets the next data item)
```

What we have done is to take a substring from the C$ string containing the data items.

---

TRUNCATE:

To convert the following functions LEFT$, RIGHT$ MID$, TL$ from TRS-80 to Sinclair.

Use the following conversions:

| TRS-80 | SINCLAIR |
|---|---|
| LEFT$(A$,N) | A$(1 TO N) OR A$(TO N) |
| RIGHT$(A$,N) | A$(LEN A$-N+1 TO) or A$(LEN A$-N+1 TO LEN A$) |
| MID$(A$,M,N) | A$(M M+N-1) |
| TL$(A$) | A$(2 TO LEN A$) or A$(2 TO) |

In all of the above cases, what we are doing is simply slicing a string expression mathamatically. I believe it is known in the manual as slicing or truncating. In LEFT$ conversion this returns the first (N) characters, starting at the first character from the string A$. A study of the other conversions will show a similar procedure.

JUMPS - Converting the ON GO TO command:

Conversions for the ON-GO TO statement.
The ON GOTO statement in the TRS-80 is in the form: ON (algebraic expression) GOTO N1....Nn , where N1 to Nm are a list of program line numbers. The expression is evaluated, control of the program continues at the line number whose position on the list is the integer of the expression.
TRS-80 example...

```
20 ON X GOTO 50,60,70  (IF X=1 JUMP IS TO LINE 50,
IF X=2 JUMP IS TO LINE 60, IF X=3 JUMP TO LINE 70)
```

Conversion for Sinclair is:

```
20 GOTO (50 AND X=1)+(60 AND X=2)+(70 AND X=3)
```

Type in exactly as written, the AND is ANDing a memory value and is necessary. 50, 60, and 70 are the lines jumped to.....change them to suit your program.

## MTERM TIPS

**I've noticed a lot of M-TERM tips and thought I'd list a few:**

**#1:**** TO SET M-TERM RAMTOP******

**When you enter M-TERM with a PRINT USR 54016 it CLEARS and sets RAMTOP at 54015. To CHANGE THIS just POKE 54024,103 then POKE 54025,191.**

**#2:****** TO CHANGE BUFTOT********

POKE 61166,70 and POKE 61167, 190 The two byte number you want to get is a number that WHEN 26710 is SUBTRACTED from it, will give the answer of what you want your BUFTOT to be. It is a good idea to make it a couple of hundred bytes LESS than the actual space that's available so that there's WORK-SPACE ROOM to use when you exit to BASIC and save to disk or tape. It stops the worrying about WHEN to save. Just wait until the "buffer full" signal..

NOTE** THE ABOVE POKE VALUES ARE FOR A RAMEX DISK "MDOS;0" BUT... THEY WILL WORK IN ANY SITUATION

#3:***** TO AUTO-RUN RAMEX "MDOS;0" ******

Usually, when you initialize the "MDOS;0" you're returned to Copyright Message and have to enter your next command. To get past this, do PRINT #4: LOAD "MDOS80" CODE:: POKE 49647,201: RANDOMISE USR 49644. The "201" poke makes it possible to do other commands after initializing..like loading "M-TERM"CODE: PRINT USR 54016 and then maybe load your REPEAT DIALER program or whatever.

#4: READ SAVED FILES WITH A LOOP

You don't need to have M-TERM loaded in order to read files... A loop like FOR J=26710 to 48999: PRINT CHR$ PEEK J;: NEXT J: will do it. The same loop with this addition..IF CHAR$ PEEK J = 13 THEN LET J = 8 + 6 (before the PRINT) is handy to check on how you are doing when preparing FILES with LINE #'s and REM's but you have to change the loop start to 26715 (or you'll be off by six) I hope something here will be of help to you....

Downloaded from Compuserve Data Library August by Greg Lloyd

## SUPER DATA SAVE

The SDS filter and software are designed to be used with the ZX81 TS1000 to give a save load speed of 16K in 29 seconds. In addition to this it verifies saves, saves loads data blocks only, loads named programmes or data blocks, scans a tape for named programmes or data, gives bytes free and renumbers basic programmes.

## SDS FILTER ASSEMBLY INSTRUCTIONS

Install all resistors, then capacitors, then diodes then IC's. Install a jumper in the position marked J.

Wire a 3.5mm jack to the position marked E, in the normally closed position, for the monitoring earphone. Wire a 3.5mm jack, in the normally closed position, for the 9V Timex/Sinclair power supply.

Modify the cassette case by taking apart and removing the two protrusions that keep the cassette in place. Drill a hole in each end 20mm from the edge that opens so that the jacks can be installed. Drill a 3mm hole 30mm from the edge on the right end. Run a wire with 3.5mm plug through the hole just drilled and wire to the 9V points on the PCB. Run jumpers from these points to the 9V DC output jack observing proper polarity. Install the jack on the right side of the case. Run a wire with a 3.5mm plug through the computer ear hole in the case and install to the appropriate points. Run a wire with a 3.5mm plug through the cassette ear hole in the left side of the case and wire to the appropriate points on the PCB. Run a wire with a 3.5mm plug through the computer ear hole in the right side of the case and wire to the appropriate points on the PCB. Install the jacks in the case.

Test the unit. If any problems check all soldering connections, wiring and component polarity and positioning. Especially check the two diodes at the filter output as these may blow if the filter is powered down before removing from the EAR jack on the computer. Insert the PCB in the case along with the label. Reassemble the case and seal shut.



## PARTS LIST FOR SDS FILTER

Resistors 1/4watt

R1 100K (Brown, Black, Yellow) 3 each
R2 10K (Brown, Black, Orange)
R3 120K (Brown, Red, Yellow) 2 each
R4 24K (Red, Yellow, Orange)
R5 5.6K (Green, Blue, Red)
R6 1 Meg (Brown, Black, Green)
R7 22K (Red, Red, Orange)
R8 4.7K (Yellow, Violet, Red)
R9 20K (Red, Black, Orange)

Capacitors as small as possible

C1 10uf 25V Electrolytic 2 each
C2 22uf 16V Electrolytic Axial
C3 0.1uf (0.1Z) 2 each
C4 0.001uf Tantalum +/- 5% (1000G) 2 each
C5 0.047uf (473 or .047M)
C6 0.022uf (223Z or 0.022)
C7 68pf Tantalum +/- 5% (680K or 68P)

IC's and Diodes

IC1 555 8 pin
IC2 4136
D1 1N4148 diode 6 ea

Miscellaneous

Jumper wires for PCB
3.5mm panel mount mono earphone jack 3 each
3.5mm mono earphone plug on 12"-16" cord
Cassette case
Label
Circuit Board
SDS software and manual (included with system only)

Circuit Boards are available from Integrated Data Systems, 30 Brookmount Rd. Toronto M4L 3N1 for $10 + PST + $1.00 P&H. The complete system is $50 + PST + $1.50 P&H. Offer expires 11/30/86.

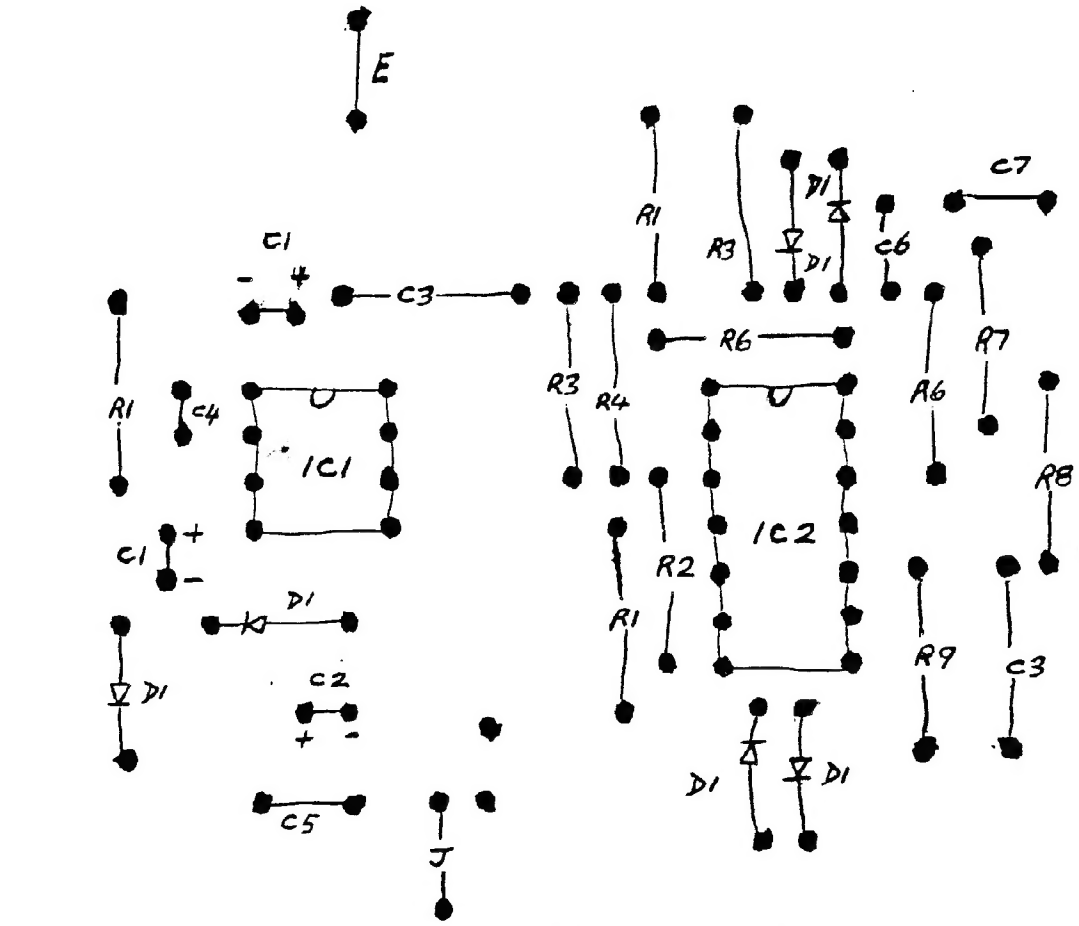
## TIMEX 2068 REPAIRS

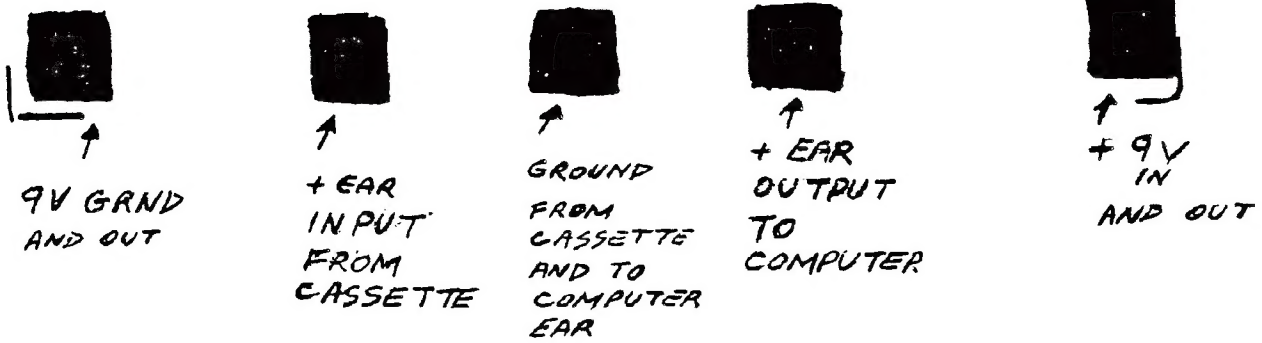
Quotation from a letter I received from one of our our out of town members, Vlad Trcka.

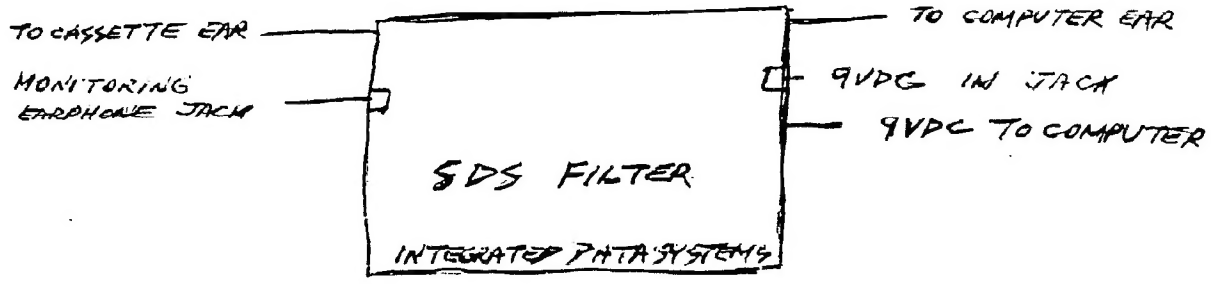
"To describe my odyssey with the computer repair: I sent the computer to Arkansas on June 6, 86 and I enclosed a money order for $30 (US). The computer was in it's original box, minus the ROM chip, which survived the mishap (unplugging the DD), so I kept it. I received acknowledgement from Timex on July 18,86. Before I mailed the computer to Timex I took it to Customs Office and I had the serial number registered. The computer came back finally on Aug. 25, 86. I had to pick it up from Customs, and from the amount spent for the repair, $29.95 (US) I paid duty and Federal tax - $6.76. The computer looks brand new (probably is) and the box contained everything that comes with a new computer: book, power adapter, cords, and the TV-Computer switch (hook-up)"

G. Chambers.

SDS PCB PARTS LAYOUT (COMPONENT SIDE)



# ARTIFICIAL INTELLIGENCE

It seems that I have always been interested in "the study of how computers can be made to act intelligently," or in other words, Artificial Intelligence. But I suppose I really became interested in Artificial Intelligence, or AI for short, when I purchased my first computer, a Sinclair ZX81. I was fascinated by the way a computer could solve problems and follow instructions so precisely. However, I find myself becoming increasingly dissatisfied with all the new "high-tech" computer equipment I find myself surrounded by each day. What I am talking about is the problem of computer-programmers-headache. This is similar to "tennis elbow", or a runner's "athletes foot". It's not a new dilemma, but it's an annoying one. No matter how advanced the technology is that we work with, there are always some tasks that become tedious with time. There are many ways in which smart AI systems can be implemented to make life more enjoyable for everyone. This technology should not be feared, computers will always be under the control of man. Even though man will leave many tasks to machines to think about and perform, he will still have many tasks to perform that can't be carried out by any computer or machine.

One of the major problems with computers today is that they are stupid. They have to be reprogrammed to perform even the simplest new task. It would be ideal if someone could write a very smart program that could program other computers. The development and application of an AI system that could program computers is one of the most difficult, and fascinating projects facing computer scientists in this decade.

The development of an AI system would revolutionize automation.This type of automation would not be physical, but mental. In the near future, processes that require a logical examination of many facts, and require split second decisions, will probably be automated by an AI system. We might find this type of AI on assembly lines of the future. Consider this scenario: A GM car assembly line experiences a short circuit in one of its' robot workers. As a result plastic sheilding on a high voltage wire nearby is accidentally ripped and left bare by the malfunctioning robotic arm. To make matters worse, the resulting sparks have started a fire in the spray painting section, setting off the fire-warning-system. Instead of allowing thousands of water sprinklers to flood the fire and the live wires, the intelligent assembly-line-monitor, directs an alternate course of action, far less dangerous than the former. It sounds the evacuation alarm. When all human workers have exited, it locks all human accessible doorways, sprays the electrical fire started by the live wires with CO2, and uses highly controlled water jets

to extinguish the solvent fire nearby. After circuit and atmosphere tests show that no danger to humans remains, the doors are unlocked. Later, a complete emergency-diagnostic report is printed and filed, in unvolatile memory, for a permanent record of the accident.

Fantasy? No, most of the technology described above is available now. In fact, systems similar to this may even be in existence today.

I hope that this look at the advantages, and potential of Artificial Intelligence has given a greater insight into the current and future state of technology. We will benefit greatly from AI technology. We should not be afraid to leave a few, or many, decisions to machines if we have faith in our own intelligence and ability.

* Author:Unknown
* Downloaded from The Ascii Exchange

SOFTWARE REVIEW

Game: BOMBJACK Rating 9.5/10

T/S 2068 with a Spectrum ROM catridge

BOMBJACK is one of the best games of 1986. The graphics are the best I have seen. In BOMBJACK you are flying around the screen collecting bombs. You have to collect all the bombs before you go on to the next screen. There are flying birds that can kill you. Robots turn to flying saucers when they reach the bottom of the screen.

After collecting 11 bombs a flying "P" comes out. After getting the "P" the birds and robots freeze up and you will be able to kill them.

Once in a while a flying "B" comes out which means bonus, and a flying "E" comes out which gives you 500 points and an extra Man to play.

This game is very addictive. Sound effects are great, too.

by Renato Zannese

LARKEN DISK DRIVES

by Peter Hacksel
247 Queen St. W.,
Toronto Ont . M5V 1Z5
(416) 596-1-663
Compuserve 73537, 1436

After a four month leave of absence from the land of computers, (And land itself for that matter!) I have returned and so plan to resume my column. In my last article I showed how to set up the Larken Drive. Now that you have your computer set up and having read the manual you now should be enjoying your drive. The Larken Drive is as we know somewhat limited in it's range of commands as compared to other drives. For example the Larken DOS does not support commands for such things as Random Filing, copy disk, etc. However there are ways of getting around such things. All it involves is a little understanding of how the LARKEN DOS stores it's information on the disk.

For those who don't already know, a disk drive is something like a cross between a record player and a tape recorder. The disk is made of a substance much like what is in a cassette tape however it is capable of storing a larger amount of information in a smaller area. The disk resembles a record in that the information is stored in a circular fashion and is read by spinning the disk and placing the head of the disk drive onto the disk. The major difference between a record and a disk is that a record has one groove spiralling in toward the center label whereas the disk has a total of 80 tracks (40 on each side.) These tracks are separate rings and do not spiral.

Each track is labeled from 0-79 and may contain a total of 1960 bytes of information. The drive has a head on each side of the disk which moves back and forth across the disk by jumping from track to track.

Now to keep track (no pun intended!) of all this information the DOS uses the first track (Track 0) as a directory. On this track is stored a list of all the programs on the disk and where they may be found.

For example let us say that you want to save a program that is 500 bytes long on a newly formatted disk. First of all the DOS will check to see if there are any programs stored on the disk by looking in the directory. It finds that Track 1 is empty and saves the program onto that track. the DOS then records in the directory the name of the program and the track to which it was saved. (in this example Track 1). Now although the program only used 500 bytes of the 1960 bytes available on the track , the entire track is reserved for that program and no other information may be stored there unless that program is deleted.Now let us say that we would like to save another program which is 6500 bytes in length. This will require 4 tracks to save it.(We say that the program is 4 Blocks long) For this program the DOS will check the

## USEFUL ROM ROUTINES IN THE TS 2068

RST 8 (or CALL 8)--Report an error.

This causes the routine to halt, and report the error whose code is one MORE than the following byte. E.g. RST 8 DEFB 8 is a STOP (Error 9). DEFB 9 for ERROR A, etc. Useful for programming your own error reports.

RST (or CALL) 10H--Print CHR$(A)

Prints the character whose code is in A to the current output stream. This can be used with all the "control characters" listed in the character set, followed by loading any necessary operand or operands into A and using another RST 10. I found that TAB requires two operands like AT, even though the second makes no difference!

CALL 436 H--BEEP

To do a BEEP X,Y, X and Y must have been sucessively placed on the calculator stack with the routines for that purpose. E.g. BEEP 1,30 LD A,1 CALL 30E6 LD A,30 CALL 30E6 CALL 0436H

CALL 73F--PRINT MESSAGE

A very useful routine for handling "ragged" message tables. The table should be at address TABLE, starting with a byte 80H, and with the last character of each message "inverted" by adding 128 to its code to mark the end. The routine is called with TABLE in DE, and the message number in A (The first message is "0").

CALL 8A6--CLS
CALL 8A9--CLEAR LOWER SCREEN
CALL 939H--SCROLL
CALL 938--PART SCROLL

Scrolls the bottom "B" lines, leaving the screen above intact.

CALL 97F--PART CLEAR

Similar to the last, but CLEARS the bottom B lines.

CALL A 02--COPY
CALL DID--NEW
CALL 1230H--OPEN CHANNEL

Load into A:1 for lower screen, 2 for main screen, or 3 for printer and call this routine to direct subsequent output. Unless a PRINT or LPRINT statement is in the same BASIC line preceding the USR call, the lower screen is the "default channel" until another one is opened by this CALL.

CALL 26EE--DRAW X, Y,Z

The parameters must be stacked, and H'L' saved as for CIRCLE.

CALL 2813--DRAW X,Y

CALL with ABS X in C, ABS Y in B, SGN X in E, and SGN Y in D; and save H'L' as above.

CALL 2E70--STACK STRING

Passes the "parameters" of a string to the calculator stack as an argument for string functions. At entry, A holds 1 if the string is "simple", 0 if it is DIMensioned or a slice; DE holds the string address; and BC holds the length.

CALL 2 EB2--MULTIPLY 16

This multiplies HL by DE, giving "Out of Memory" if the result overflows.

CALL 2FAF--UNSTACK STRING

Retrieves the "parameters" of a string result from the calculator stack; address to DE, length to BC.

CALL 3046H--ALPHANUM

Returns with CARRY if A holds the code of a letter or digit, NO CARRY otherwise.

CALL 3048--ALPHA

Returns CARRY if A holds a valid letter code.

CALL 30D9--NUMERIC

Returns with NO CARRY if CODE "0"<= A>=CODE "9".

CALL 30E6--STACK

A Converts the integer value in A to floating-point and puts in on the calculator stack.

CALL 30E9--STACK

BC Converts the integer value in BC to floating-point and puts in on the calculator stack.

CALL 3160H--UNSTACK BC

Retrieves the top value from the calculator stack and loads it into BC

CALL 31A1--PRINT FP

Prints the top number on the calculator stack. The use of the "floating-point calculator" (RST 40) is "beyond the scope of this article." Hopefully there will be another piece, after I test some of the more obscure functions.

Wm. Linden

directory and find that Track 1 has been used and that the next 4 availiable tracks are 2,3,4 and 5. The program is then saved to these tracks and it is noted in the directory that these tracks are reserved for the second program. When it comes time to load a program all the DOS does is simply move the head to Track 0, read the directory, find which tracks contain the program and then move the head over to the tracks and one by one read the program into the computer memory. If you decide that you wish to delete a program the DOS simply erases the name of that program from the directory. This means that if you try to load that program again the DOS will look in the directory and will not find the name of the program and assume that there is no program with that name on the disk and will return with a "NO FILE" error report. NOTE: When a program is deleted the name is removed from the directory but the data is NOT erased. This means that it is possible to recover this data provided that you do not save another program overtop of the data. This means that if you accidently delete an important program it is possible to recover the lost data. More on this later.

This should give you a very basic understanding of how the information is stored on a disk. With my next article we will investigate ways of using routines from the DOS to manipulate the data and increase the powers of the drive enormously.

## SABREWULF

Some useful POKEs for the game SABREWULF:

Here is how to add these POKES:

1. Load the first part of Sabrewulf, stop tape and press break.
2. Type:
   POKE 23756,1:CLEAR 65535
3. Edit the line, move the cursor to the end and delete the
   PRINT USR 23424.
4. Add line 10 with any of the following POKEs:

   POKE 43575, 255 for infinite lives, one player.
   POKE 45520, 255 for infinite lives, two players
   POKE 45599, 255 for number of initial lives - 1 to 255.
   POKE 41725, 255 for no limit on gained lives

5. Next add line 20:
   20 PRINT USR 23424
   and finally Run to load in the rest.

from the Oct. 1984 issue of YOUR COMPUTER - page 36

## THE FUNCTION DISPATCHER

by Mike Lemyre

The routine to call the function dispatcher i found on pages 214-216 of the Timex Sinclair 206 Intermediate Advanced Guide Some people have bee unsucessful in finding a use for this utility. Howeve I was reading an article in the news letter of th Personal Computer Club of Toronto; which dealt wit the I.B.M. function dispatcher when calling D.O.S functions I noticed a similiarity between the tw utilities. The priciples to keep in mind are:

1. Initialize machine stack.
2. Push service code.
3. Set up registers to function requirements.
4. Call the dispatcher.

Compare these routines.

| I.B.M. | TS2068 |
|---|---|
| xor ax,ax | LD DE 0000 |
| push dx | push DE |
| push ax | push DE |
| mov ah,2h | ld DE 135 |
| mov dl,38 | push DE |
| int 21h | ld A,38 |
| ret | call 6200 |
| | ret |

Both of these routines print the & character. Som other routines that have been sucessfully called are:

| Service Code | Function |
|---|---|
| 34 | cls |
| 89 | plot bc |
| 135 | print char |
| 141 | cls |
| 142 | scroll |

I will expand on the uses of subroutines an function calls in my next article.

## ERRATUM

Disk Droppings

In the last issue of the newsletter (4.4) there were a couple of editting errors in the listings in Greg Lloyd's article, "Disk Droppings". The two listings below are the corrected versions.

```
10>LET i=33280
 20 IF PEEK i<32 THEN  POKE i,32
: LET i=i+1: POKE i,32: GO TO 20

 30 IF i>52480 THEN  STOP
 40 LET i=i+1: GO TO 20
```

```
10>LET i=33280
20 FOR k=0 TO 19199
30 IF PEEK (i+k)<32 THEN  POKE
(i+k),32: POKE (i+k+1),32
40 NEXT k
```

CALL 133F--CLEAR EDIT LINE
CALL 134E--CLEAR WORK SPACE
CALL 1354H--CLEAR CALCULATOR STACK
CALL 1541H--LLIST
CALL 1545H--LIST
CALL 15A1--PRINT LINE
Print the program line whose ADDRESS is in H L.

CALL 16D6--LINE ADDRESS
Enter with a program line number in HL. Returns line address in HL.

CALL 1745H--DIFFERENCE
Loads BC with HL minus DE, and switches HL with DE. Useful for finding string lengths.

CALL 1788H--PRINT NUMBER
Print the number in BC.

CALL 1ED4--RANDOMIZE
CALL 1FF2--PAUSE
Pause for BC ticks.

CALL 21DB--PRINT STRING
Print to the current channel a string of length BC at address DE.

CALL 2603--PIXEL ADDRESS
Enter with C holding the graphics X coordinate and B holding Y coordinate . Returns with HL holding the address of the appropriate display file byte, and A holding the bit number corresponding to that pixel.

CALL 263E--PLOT C,B
Load X coordinate into C and Y into B to do a PLOT.

CALL 266D-- UNSTACK A
Retrieve top value from calculator stack to A.

CALL 2686H--CIRCLE
To execute CIRCLE X,Y ,Z, the three parameters must be sucessively placed on the calculator stack. Also, the routine corrupts H'L', which is needed for a return from a USR. Thus, it has to be used:

```
 EXX
PUSH HL
EXX
LD A, X
CALL 30E6
LD A,Y
CALL 30E6
LD A,RADIUS
CALL 30E6
CALL 2686; CIRCLE X,Y,RADIUS
EXX
POP HL
EXX
```

## QSPELL - A Review
by Larry Sadler

The following is a quick review of QSPELL, the spell-check program obtained from Curry Computer in Glendale, Arizona 85312-5607.

QSPELL is a program that has a 25,000 word dictionary, with room to add an additional 1,000 words. It has features that allow you to use it to solve crossword puzzles, anagrams, or other word puzzles that make many words from the letters of a given word. It has patches to hook into Quill so that the output of QSPELL can be exported to Quill for correction.

I do not recommend QSPELL for routine document writing. It is not an integral part of Quill. You write the document, save to cartridge, load the QSPELL, load the saved file, and choose between ignoring, marking, or adding each word of the document that is not in the default dictionary. QSPELL then creates a new file as named, which is then reloaded into the reloaded Quill. The new document has scrambled the words marked by the dictionary so that you can delete them and insert the correct word (if you can remember what it was!).

The merit of QSPELL is it's game and puzzle features, and that it can be used to check the variables in a program. Another advantage, for some, is the fact that you can create customized dictionaries of your own - e.g., jargon for a particular industry (computer, electronics, medicine, engineering, etc.).

The demerit is that QSPELL is separate from Quill and does not provide for editing/correcting from within Quill. I suspect that this is because the 128K of standard RAM is not sufficient to accomodate a version of Quill that integrates a dictionary feature (or a Theraurus). The dictionary alone occupies 70K. Perhaps when the QL becomes standard with,say, 640K or so RAM, a new version of Quill will be produced. Till them get a second opinion on your work before you send it out.

EXPLOSION:

```
        10 SOUND 6,6;7,7;8,16;9,16;10,16;12,56;13,8
        20 PAUSE 90
        30 SOUND 8,0;9,0;10,0
```

Dear Out-of-town members,

SEPT 14/1986

Every time that I sit down to compose this column I hope to be struck with an inspiration. It never happens and this time is no exception.

We had a sort of a delay in the publishing of our newsletter during the past several months. However we now have a push on with a new editor and we expect that we will have two more issues out by the beginning of December. That will put us back on track as far as 6 issues a year goes. Fortunately we have several regular columnists who contribute a page, and that makes the whole task really quite easy. I pasted up the MAY/JUNE issue, and the new editor and myself put out this issue. It is only 12 pages this time because a couple of columns did not appear on time. However I expect that we will have no trouble in filling 14 pages henceforth.

Quite a bit of our letter is filled with LARKEN stuff. I suppose this is natural because the more active members are often the ones who have the Drive, and they naturally want to write about it. Mostly, the disk systems that are offered for the Timex are really just a barebones DOS, without any particular utilities to take advantage of it's possibilities. The LARKEN system is no exception. The difference is that with so many active members in our club owning it, we are developing quite a suite of utilities to go with it.

There is a disk DOCTOR program which enables one to look at the track makeup and to readily make changes on an individual bit basis. Then there is a program to facilitate the copying of programs on a single drive system. Bob Mitchell has written a program to make it easy to change the name of a program on a disk; also another program to maintain an index of programs that one has on disk. Altogether a very exciting time. I am trying to pursuade someone to write a program to be used with the MTERM II program to allow for the automatic transfer to disk of the MTERM data buffer after it is filled up. This would allow for the collection of any amount of data via a modem. Just insert a new disk, when that one is filled up. Well, you see what I mean.

One other thing. I see where I made a typo error in the last issue, where I said that I could hold 5 files on one disk, with the LARKEN. Larry Crawford, one of our members from London,Ont., took me up on that saying that that wasn't particularly good; he could hold 8 files on his (OLIGER) system. I meant to say 50 files, that is I have four files used for the Tasword program and 50 letter headers, that I call up when I address a letter to an out of town member. That's 54 files total. Sorry about that, Larry; I love my LARKEN!!

You may remember me telling you about getting some printer ribbon ink,and starting to ink my own used ribbons. Well, it is working out pretty well. I am sometimes a little too generous with the ink on the ribbon, but it is really going not too badly. They supply me with a pair of plastic gloves to handle the ink. I haven't used them, but I must say I should. Pretty hard not to get ink on the fingers.

In the current newsletter there are a couple of comments about how disk prices are coming down. I bought a package of 10 for $8.28, plus PST a few days ago and this is better that either Ian Robertson or Greg Lloyd reported in their columns. I bought them from INTERNATIONAL MARKETING CORPORATION, 133 Midwest Road, Scarborough. (416) 759 6721.

I have put together several folders containing material on different topics. There are binders for LARKEN, MODEMS, and Hacking, so far. If anyone is interested in borrowing them just drop a line. You will have to ask you to return them quite promptly, because they are proving to be popular among our in-town members, but you should be able to Xerox any items that you are interested in, and return it.

I'm not sure whether you read the item in the newspapers quite recently about a new computer being testmarketed in the USA. It is being produced by Hyundai. Yes that's right, the auto-maker (I kid you not, to quote Jack Carson). It was in a business column. It is to be a sort of IBM clone, but cheaper than the other clone competitors. Being handled in the USA by the firm, BLUE CHIP ELECTRONICS, and sold through discount stores.

I have mentioned to some members that I have made up several copies of tapes that I call "Games Tapes". That is what they are; games for the TS2068. There are 3 tapes; tapes 1 and 2 are for the Spectrum or the TS2068 with a Spectrum ROM; the other tape 3 is for the TS2068. Ask for one of them, if you are interested. I would ask that you not keep them for too long though.

I mentioned in an earlier letter that our treasurer had suffered a heart attack. He has since had an operation, which was successful, and is due to go back to work in October, I believe. We have another member who has had the same kind of problem. Anthony Youatt, from Nipawin, Sask. had much the same trouble, a heart attack, and he was to have been operated on about a month ago. I have not heard from him for some time, but I must tell you a little story, which left me feeling mortified.

Anthony called me one day, from Winnipeg, I think it was. Anyway my opening remark was," I haven't heard from you for so long, I thought you'd died." Forgetting that he had written to me a while before, telling of his heart attack. I must stop using that comment. It was alright when I was younger, but it's getting too close to the bone now!!

For those of you who are able to make it to Toronto you may be interested in a computer show to be held at Harbourfront on Saturday, October 18th from 10 a.m. to 6.p.m. It is billed as "the computer show for the rest of us", home users, small businesses, teachers, students, hobbyists. Cost is $4. admission. I don't think our club will be represented; they want $40 for a booth, and that seems a bit much. We shall see what the membership thinks. But it is the sort of show for hobbyist types.

Some of you may be wondering what the going price of a 2068 is these days. Well, I have been instrumental in getting buyer/seller together on a couple of occasions lately, where the selling price was $150. For our US members this is in $CAN.

One of our former members has given me a number of old magazines. Seven issues of SYNC, 6 issues of ZX COMPUTING. I also have the seven issues of TIMEX/ SINCLAIR USER, which I am more relaxed about loaning now. So if you are interested in borrowing a batch of them , say the word. Of course it goes without saying that you would pay the postage (expensive!); and I would like them back! I also have all the issues of YOUR COMPUTER magazine. I think I would send them 4th class, and that is quite a bit cheaper.

One of our members, Eric Michaud, 1269 Andrew Court, Sarnia, ONt. has built a digitizer for the 2068. With it he is able to pick up images from the output of another computer, a VCR or TV and capture/display them on the 2068. He uses the joystick port as the input connection. I mention this project of Ian's, because it is so fascinating, and because he would be interested in hearing from others with a like interest. So do drop him a line if you are interested. He has sent me some of the images on a LARKEN disk. I made prints of them on My 2040 printer and took them to the club where they created considerable interest.

Anyone know of a CHEAP video camera? Ian is looking for one for an extension to his project.

Do any of you out there have a pottery friend. I mean someone who mixes their own glazes, etc. I ask this because when I first bought my ZX81 I learned programming by writing a masterpiece program to perform glaze calculations. Funny thing though, having done that, the program just languishes there because I know practically no potters to have them use it and marvel over it. Really I am dying to get it into the light of day, so if you know of someone who would be interested in it, get us together. It was designed for the ZX81, and adapted for the 2068. Actually it is in the club library too, but I think computer nuts don't get turned on by pottery making.
(I used to throw clay, until I became president of this club; now I don't have time for it!!)

Another of our members has sent me a letter which I understand was produced with a SHARP CE-515P four-colour Printer Plotter. I mention this because he mentions that they are on sale at Total Office Systems, Unit 14, 1050 McNicoll Ave. Scarborough (493 3575, for $99. Roy writes," I find it to be excellent for plotting and for the few letters I write."
And the price certianly seems right!

You might take note that Consumers Distributing have the COMPUDECK tape recorder in their current catalog. It sells for $33, and is very highly recommended by all of our club members, and the price is right also. It is not suitable as a normal recorder, because there is essentially no loudspeaker, in the normal sense.

One of our intown members, in fact it is our newsletter editor,Sean Wenzel, is setting up a bulletin board. It is almost ready to go; he is just waiting for a phone line. I hope that we can take advantage of it by creating a TIMEX section on it. I cannot tell you anymore about it yet but I hope that the next newsletter will have lots of information about it. It is not on a 2068, rather, I think it is an IBM clone, but what the hell, we aren't fussy, are we! We'll take it.

Sincerely,
George C.